

Internacionalização em PHP

Este artigo tem como objetivo cobrir o básico da internacionalização de aplicações em PHP. Atualmente, com a grande demanda de desenvolvimento software livre, surge e necessidade de distribuição das aplicações com possibilidade de rodar em vários idiomas.

Introdução

O gettext é usado no linux para internacionalização de programas em C, C++, shell script, python, perl e PHP. Com a biblioteca GNU GETTEXT, uma aplicação poderá ser utilizada em vários idiomas.

O gettext dispõe de utilitários que extraem strings **marcadas** no código fonte da aplicação para um arquivo de texto (extensão .po), a partir da qual um tradutor faz as devidas traduções em suas língua. O programa tem a responsabilidade de verificar o locales do ambiente, carregar a biblioteca gettext e as strings traduzidas de acordo com o idioma configurado.

O manual on-line do gettext poderá ser encontrado em:

<http://www.gnu.org/software/gettext/manual/gettext.html>

Cuidados especiais devem ser tomados com relação à codificação dos arquivos.

Instalação de pacotes básicos

Para usar estas funções você deve baixar e instalar o pacote do GNU gettext em »

<http://www.gnu.org/software/gettext/gettext.html>

ou, para o Debian e similares, instalar com apt-get

```
# apt-get install gettext
```

Configuração do PHP

Habilitar no arquivo php.ini o módulo gettext.so. Adicione uma linha na área **extensions** conforme abaixo:

```
extension=gettext.so
```

Confirme o caminho das extensões do PHP (arquivo PHP.INI), adicionando o path abaixo:

```
extension_dir = "/usr/local/php5/lib/php/extensions/debug-zts-20060613"
```

Salve o arquivo e reinicie o Apache.

Caso o PHP tenha sido instalado através de compilação, certificar-se que o PHP seja compilado com a opção abaixo:

```
--with-gettext=shared
```

Implementação

O funcionamento básico dá-se pela substituição de todos os strings da aplicação que são enviados para a tela. No PHP, isto se dá pela utilização do comando **echo** por exemplo.

Exemplo:

- Impressão normal:

```
echo ("Caminho de casa");
```

- Impressão com gettext:

```
echo gettext("Caminho de casa");
```

Ou seja, utilizar a função gettext sempre que for enviado algo para tela.

- Exemplo:

No exemplo abaixo, o texto "Caminho de casa" será impresso em Espanhol: "Camino de la casa"

```
<?php
$language = 'es_ES.UTF-8';
//configura a variável de ambiente LANG para es_ES
putenv("LANG=$language");
// configura a variável de localidade para es_ES
setlocale(LC_ALL, $language);

// Messages é o domínio do arquivo de mensagens
$domain = 'messages';
bindtextdomain($domain, "/var/www/apache2-default/inter/locale/");
textdomain($domain);

//O texto abaixo sera traduzido para o espanhol.
echo gettext("Caminho de casa");
?>
```

Acesse o diretório da aplicação WEB e digite o seguinte

```
# xgettext -n *.php --language=PHP
```

obs.: Verifique a codificação do seus arquivos PHP. É provável que você deverá especificar o tipo de codificação.

Ex.:

```
# xgettext -n *.php --language=PHP --from-code=ISO-8859-1
```

ou

```
# xgettext -n *.php --language=PHP --from-code=UTF-8
```

Obs.: Você poderá criar um arquivo messages.po para cada página de sua aplicação. Para isso, substitua o caractere curinga (*) do comando gettext pelo nome de seu arquivo PHP.

Ex.:

```
# xgettext -n getmessages.php --language=PHP
```

O comando `xgettext` irá extrair as strings de todos os arquivos PHP do diretório atual e irá gerar um arquivo **messages.po**, que contém a tradução das strings do seu código. No nosso caso, conterá a tradução do string "Caminho de casa".

Na última linha do script, temos o código `echo gettext("Caminho de casa");`, que indica ao `gettext` que o conteúdo "Caminho de casa" deverá ser traduzido. Isto trata-se de uma marcação, indicando que aquele string deve ser traduzido.

Será gerado um arquivo com extensão **.po** no diretório atual.

Crie um diretório `locales/es_ES/LC_MESSAGES`, localizado abaixo do diretório da aplicação. Este exemplo aplica-se ao idioma Espanhol.

Copiar o arquivo `.po` gerado para o diretório conforme abaixo:

```
# cp *.po /var/www/apache2-default/<sua_aplicacao>/locales/es_ES/LC_MESSAGES/
```

Edite o arquivo `messages.po` e faça as devidas traduções para as strings onde aparece `msgstr`

```
msgid "Caminho de casa"  
msgstr "Camino de la casa"
```

Altere o parâmetro **CHARSET=CHARSET** para **CHARSET=UTF-8** ou **ISO-8859-1**, conforme o caso. Isto evitará a emissão de um **Warning** informando que o **CHARSET** encontra-se definido como **CHARSET**.

Salve o arquivo

Converter o arquivo acima para o formato **.mo**, com o comando abaixo:

```
# msgfmt messages.po
```

O arquivo gerado é um binário.

Reconfigurando Locales

O script somente funciona se o seu sistema tem suporte para o locale especificado. Para `es_ES`, adicione ao seu sistema através do comando

```
# dpkg-reconfigure locales
```

Selecione `es_ES.UTF-8` e mantenha `PT_BR` como padrão (ISO-8859-1)

Feche e reinicie o Apache.

O script está pronto para ser testado através do browser.

Configuração com Arquivo .CONF

Você poderá utilizar um arquivo de configuração, que conterà a definição do idioma a ser utilizado. Este arquivo de configuração deverá ser lido pela aplicação (via require, por exemplo), e seu conteúdo deverá ser passado para a cláusula LOCALES de seu programa.

Esta estratégia é útil se você quiser ter independência do Locales. Em PHP, você poderá obter as variáveis de ambiente através da função `getenv()`, mas o retorno será o valor definido no Apache.

Exemplo de Arquivo de Configuração

Segue um exemplo simples de arquivo de configuração (em PHP)

```
<?php
//Define here your LANG
define("LANGUAGE", "pt_BR");
?>
```

Exemplo de cabeçalho de programa

O pequeno trecho abaixo é um exemplo de cabeçalho de programa que utiliza um arquivo de configuração chamado locale.conf.php

```
<?php
require("locale.conf.php");

$applanguage=LANGUAGE;

putenv("LANG=$applanguage");

setlocale(LC_ALL, $applanguage);

$domain = "getmsg";

$path_application = dirname(__FILE__);

bindtextdomain($domain, $path_application . "/locale/");

textdomain($domain);
?>
```